REMARKS

Claims 1-5, 7-12, 14-17, 19-22, 24, and 30-35 are currently pending in the present application. Support for the amendments may be found throughout the specification, such as paragraph [0052], for example. No new matter has been added by way of the amendments.

Interview Summary

Applicant's undersigned representative, Mr. Eiferman, and Examiner Robert Timblin participated in a telephonic interview on October 21, 2009, to discuss the claim amendments and remarks herein. The Examiner agreed to reevaluate the rejections in view of the amendments and remarks herein.

Claim Objections

Claim 2 stands objected to under the contention that "item" should be specified as either a "first item" or a "second item" to provide proper antecedent basis from claim 1. Claim 2 has been amended to replace "item" with "second item". Accordingly, Applicants respectfully submit that the objection to claim 2 should be withdrawn.

Claim Rejections – 35 U.S.C. § 102

Claims 1-5, 7-9, 12, 14-17, 19-22, 24, 30-32, and 35 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,029,160 to Cabrera et al. ("Cabrera"). Without conceding the merits of the rejection, Applicants have amended independent claims 1, 14, 19, and 24 to further clarify the claimed subject matter.

Claim 1 has been amended to recite, in part, a method for executing a transaction comprising at least one query language statement and at least one file system statement. Each statement relates to a user defined type ("UDT") associated with a database server. Further, claim 1 recites storing at least one field relating to the UDT in a relational database table. At least one field is a filestream field. Data for each filestream field is stored in a respective file separate from the relational database table. The method also comprises: receiving each of the at least one file system statements. Each statement comprises a call to open a first item and at least one of a call to read from the first item and to write to the first item, and a call to close the first item; and receiving each of the at least one query language statement, wherein each query language statement is associated with a second item. For each file system statement, the method includes, upon detecting a storage platform path name associated with the first item in a file system statement: forwarding the file system statement to an agent, wherein the agent performs a call to the storage platform by passing the storage platform path name to the storage platform.

Further, the claimed method comprises identifying the first item based upon the storage platform path name; passing a database engine function that returns a file system path name corresponding to the first item to the database server; performing a table look-up for the UDT associated with the first item on the database server; extracting a real file system path corresponding to the first item using the database engine function; using the real file system path to perform an operation on the first item by passing the real file system path back to the agent, wherein the agent then interacts with the file system to cause execution of file system statement. If the file system statement includes open, read and close operations, a transaction is created as part of an open operation. The transaction is managed separately from the database server and comprises the at least one query language statement and the at least one file system statement. Further, if the file system statement includes open, read and close operations, the method implements the following steps: determining whether the at least one query language statement conflicts or the at least one file system statement conflicts with another transaction; and resolving the conflict if the at least one query language statement conflicts or the at least one file system statement conflicts with another transaction. If the at least one query language statement and the at least one file system statement do not conflict with another transaction, the method implements the following steps: obtaining a read lock on a data table row associated with the associated first item for the file system statement; performing a read operation in the context of the transaction; and committing the transaction as part of a close operation.

The claim also recites that if the file system statement includes open, write and close operations, the method includes creating a transaction as part of an open operation. The transaction is managed separately from the database server and comprises the at least one query language statement and the at least one file system statement. Further, if the file system statement includes open, write and close operations, the method implements the following steps: determining whether the at least one query language statement conflicts or the at least one file system statement conflicts with another transaction; and resolving the conflict if the at least one query language statement or the at least one file system statement conflicts with another transaction. If the at least one query language statement conflicts and the at least one file system statement do not conflict with another transaction, the method implements the following steps: obtaining a write lock on a data table row associated with the associated first item for the file system statement; performing a write operation in the context of the transaction; committing the transaction as part of a close operation; and, for each query language statement, starting a transaction on the database server updating fields associated with the second item in the query language statement and sending an updategram to the database server.

Further, claim 1 has been amended to recite that ***while the write lock is obtained***: (1) ***a statement in another transaction or a non-transacted statement is prevented from accessing the row***; and (2) ***other statements within the transaction are enabled to read from the row***. Thus, claim 1 has been amended to differentiate between a transaction associated with a write lock and other transactions or a non-transacted statement. Particularly, statements in another transaction and non-transacted statements are prevented from accessing the row while the write lock is obtained, and statements within the transaction associated with the write lock are enabled to read from the row while the write lock is obtained.

For example, referring to paragraph [0052] of the present application, write locks are acquired for the lifetime of the transaction. The write lock is an exclusive lock that is acquired for the lifetime of the transaction (Present Application at ¶ [0052]). The write lock will prevent another transaction from accessing (through either read access or write access) a corresponding row while the transaction is being processed (*Id.*). The write lock will also prevent a non-transacted file system statement from accessing through either read access or write access) a corresponding row while the transaction is being processed (*Id.*). The write lock does not prevent other statements within the transaction from reading a corresponding row (*Id.*).

Cabrera does not disclose the claim 1 features of: (1) ***obtaining a write lock on a data table row associated with a file system statement associated with a transaction***; (2) ***while the write lock is obtained, preventing a statement in another transaction or a non-***

transacted statement from accessing the row; and (3) ***while the write lock is obtained, enabling other statements within the transaction to read from the row.*** The Office Action admits that Cabrera does not teach acquiring a write lock that will prevent another statement with the transaction from writing to the row (Office Action at p. 19). Further, the Office Action admits that Cabrera does not teach acquiring a lock that will enable another statement within the transaction to read from the row (*Id.*). There is no disclosure in Cabrera of differentiating between a transaction associated with a write lock and other transactions or a non-transacted statement as recited by claim 1.

Claims 14, 19, and 24 have been amended similar to claim 1 to recite that ***while the write lock is obtained***: (1) ***a statement in another transaction or a non-transacted statement is prevented from accessing the row***; and (2) ***other statements within the transaction are enabled to read from the row.*** Claims 2-5, 7-9, 15-17, 20-22, 30-32, and 35 depend upon one of claims 1, 14, 19, and 24. Therefore, for at least the same reasons set forth above, Applicants respectfully submit that Cabrera does not teach each and every feature of claims 1-5, 7-9, 12, 14-17, 19-22, 24, 30-32, and 35. Accordingly, Applicants respectfully request the withdrawal of the rejection of claims 1-5, 7-9, 12, 14-17, 19-22, 24, 30-32, and 35 under 35 U.S.C. § 102(b).

Claim Rejections – 35 U.S.C. § 103

Claims 10, 11, 33, and 34 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Cabrera in view of U.S. Patent No. 5,983,225 to Anfindsen ("Anfindsen").

Claims 10, 11, 33, and 34 depend upon one of claims 1 and 24. Therefore, for at least the same reasons set forth above, Applicants respectfully submit that Cabrera does not teach each and every feature of claims 10, 11, 33, and 34. Applicants also respectfully submit that Cabrera does not suggest the aforementioned claimed features.

Anfindsen also does not disclose or suggest the claimed features of: ***while a write lock is obtained***: (1) ***a statement in another transaction or a non-transacted statement is prevented from accessing the row***; and (2) ***other statements within the transaction are enabled to read from the row.*** The Office Action references column 13, lines 18 and 19, of Anfindsen. This context of this portion of Anfindsen teaches that "read and write modes are mutually incompatible" (Anfindsen at col. 13, ll. 8-10). Further, Anfindsen teaches that applications or transactions can specify when reading and writing is compatible (*id.* at col. 13, ll. 11 and 12). An example is provided at column 13, lines 13-19. However, there is no teaching or suggestion in Anfindsen of differentiating between a transaction associated with a write lock and other transactions or a non-transacted statement as claimed. Particularly, Anfindsen does not teach or suggest that, while a write lock is obtained, a statement in another transaction or a non-transacted statement is prevented from accessing the row, as claimed. Anfindsen also does not teach or suggest that, while a write lock is obtained, other statements within the transaction are enabled to read from the row, as claimed.

For at least the foregoing reasons, Applicants respectfully submit that Cabrera and Anfindsen, either alone or in combination, do not teach or suggest the features of claims 10, 11, 33, and 34. Accordingly, Applicants respectfully request the withdrawal of the rejection of claims 10, 11, 33, and 34 under 35 U.S.C. § 103(a).

CONCLUSION

In view of the foregoing, Applicants respectfully submit that the claims are allowable and that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested. In the event that the Examiner cannot allow the present application for any reason, the Examiner is encouraged to contact the undersigned attorney, Kenneth R. Eiferman, to discuss the resolution of any remaining issues.

Date: December 21, 2009

/Kenneth R. Eiferman/
Kenneth R. Eiferman
Registration No. 51,647

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439